



このコンテンツは公開から3年以上経過しており内容が古い可能性があります
最新情報については[サービス別資料](#)もしくはサービスのドキュメントをご確認ください

【AWS Black Belt Online Seminar】 Introduction to AWS Fargate and Amazon Elastic Container Service for Kubernetes

Keisuke Nishitani (@Keisuke69)

Senior Specialist SA Amazon Web Services Japan K.K.

24 Jul, 2018

AWS Black Belt Online Seminarとは

AWSJのTechメンバがAWSに関する様々な事を紹介するオンラインセミナーです

【火曜 12:00～13:00】

主にAWSのソリューションや業界カットでの使いどころなどを紹介(例：IoT、金融業界向け etc.)

【水曜 18:00～19:00】

主にAWSサービスの紹介やアップデートの解説(例：EC2、RDS、Lambda etc.)

※開催曜日と時間帯は変更となる場合がございます。最新の情報は下記をご確認下さい。

オンラインセミナーのスケジュール&申し込みサイト <https://aws.amazon.com/jp/about-aws/events/webinars/>

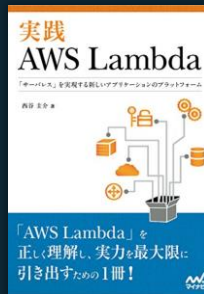
内容についての注意点

- 本資料では2018年7月24日時点のサービス内容および価格についてご説明しています。最新の情報はAWS公式ウェブサイト(<http://aws.amazon.com>)にてご確認ください。
- 資料作成には十分注意しておりますが、資料内の価格とAWS公式ウェブサイト記載の価格に相違があった場合、AWS公式ウェブサイトの価格を優先とさせていただきます。
- 価格は税抜表記となっています。日本居住者のお客様が東京リージョンを使用する場合、別途消費税をご請求させていただきます。
- AWS does not offer binding price quotes. AWS pricing is publicly available and is subject to change in accordance with the AWS Customer Agreement available at <http://aws.amazon.com/agreement/>. Any pricing information included in this document is provided only as an estimate of usage charges for AWS services based on certain information that you have provided. Monthly charges will be based on your actual use of AWS services, and may vary from the estimates provided.

Who am I ?



Keisuke Nishitani
Specialist Solutions Architect
Amazon Web Service Japan K.K



@Keisuke69



Keisuke69



Keisuke69x



Keisuke69



Keisuke69



コンテナを利用した開発の実際



そもそもなぜコンテナか？

なぜコンテナか？



パッケージング



配布



イミュータブル
インフラストラクチャ

?



結局、どういうことか？



コンテナの利点

Portability & Flexibility

Fast & Rapid

Efficient

コンテナの利点

Portability & Flexibility

Fast & Rapid

Efficient

- 再現可能な環境を自由度高く容易に定義できる
- 特定イメージはいつでもどこでも実行しても同じ環境

容易な構成管理と自動化

コンテナの利点

Portability & Flexibility

Fast & Rapid

Efficient

- 起動が速い
- 開発 - テスト - 本番まで一貫したイメージを利用したCI/CDパイプライン構築が可能

「繰り返し」に強い

コンテナの利点

Portability & Flexibility

Fast & Rapid

Efficient

- リソースの有効活用
- 弾力性の高いシステムが構築できることによる『ムダ』の削減

高いコスト効率

コントロールプレーン / データプレーン

コントロールプレーン = コンテナの管理をする場所

- どこでコンテナを動かす？ 生死は？ いつ止める？
- デプロイ時にどういう風に配置する？

→ Amazon Elastic Container Service (ECS)

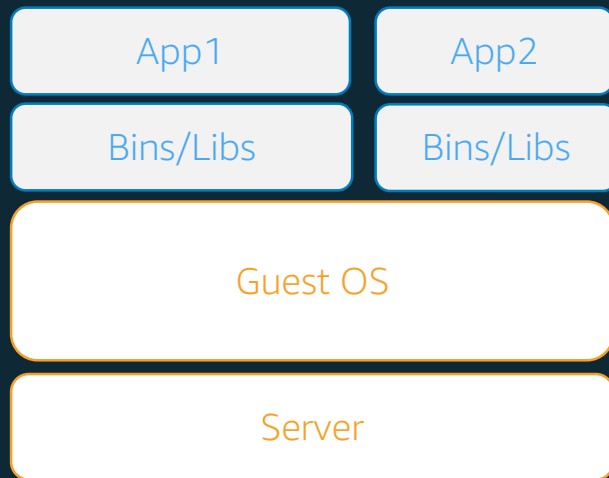
Amazon Elastic Container Service for Kubernetes (EKS)

データプレーン = 実際にコンテナが稼働する場所

- コントロールプレーンからの指示に従って起動
- 各種状態をコントロールプレーンにフィードバック

→ AWS Fargate / Amazon Elastic Compute Cloud (EC2)

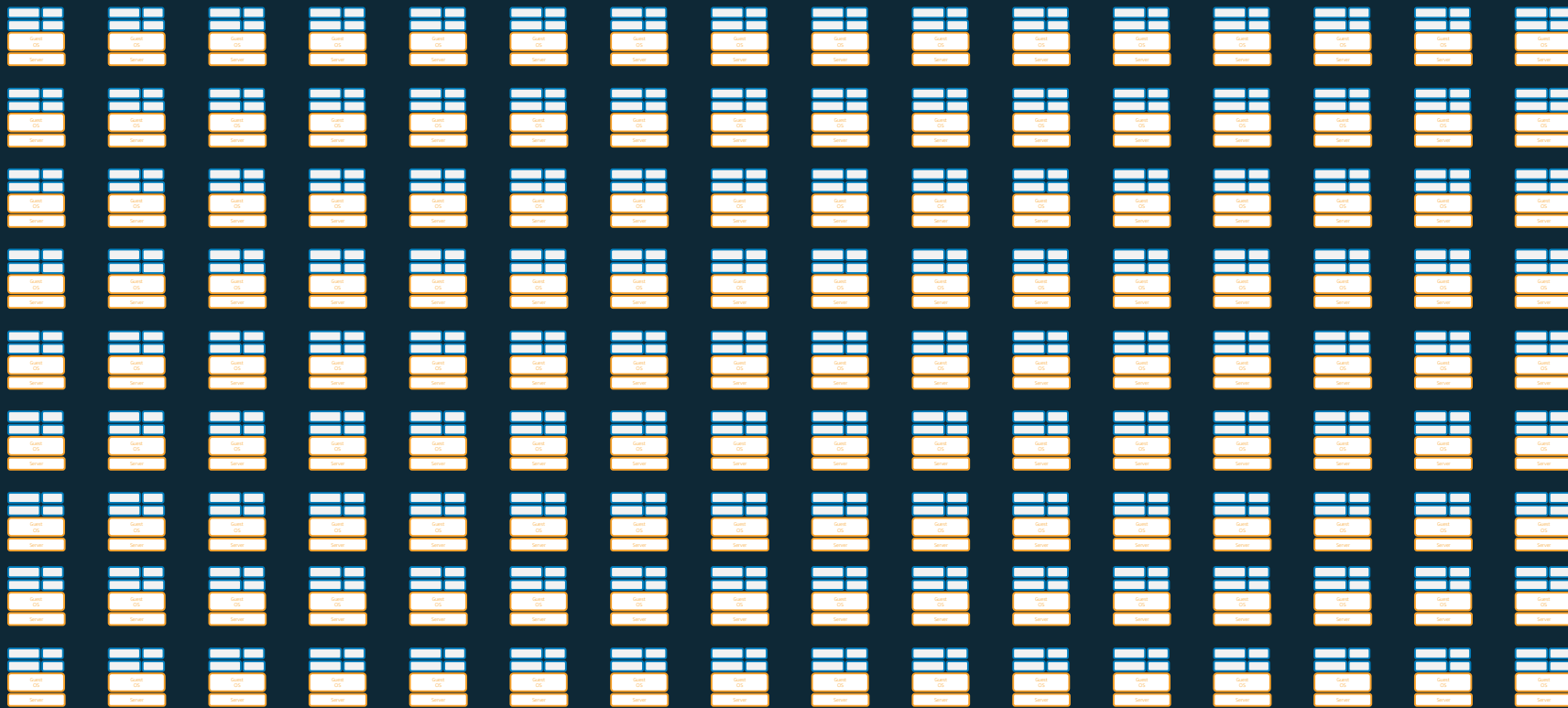
1台のサーバでDockerコンテナを使うのは簡単



サーバが増えると？



さらに増えると？





Amazon Elastic Container Service

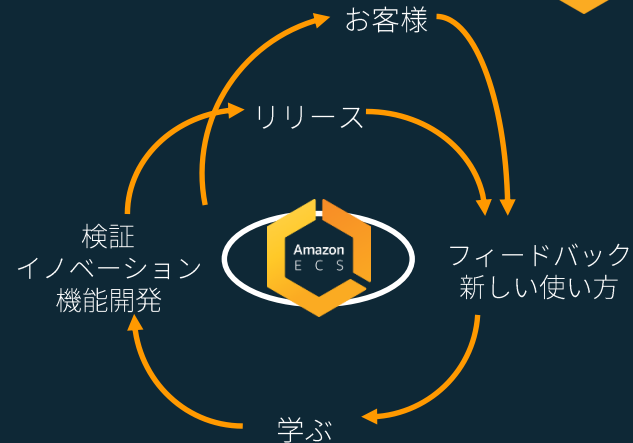


450+%
年間アクティブユーザの成長
(2016年と比較)



数億コンテナ
が毎週起動

数百万もの
インスタンス上で



50+

2015年のGA以来
リリースした機能の数

AWS上の本番環境のコンテナ運用を支援



Linux & Windows



AWS VPCネットワークモード



タスク配置



他のAWSサービスとの深い連携



ECS CLI



グローバル展開



強力なスケジューラ



オートスケーリング



CloudWatch メトリクス



ロードバランサ

その他のAmazon ECSの特徴

Amazon CloudWatch Logsと簡単に連携

Amazon CloudWatch Eventsに各種イベントが流れる

EC2へのTask配置を柔軟に設定可能

アプリケーションの開発に集中したい





AWS Fargate

新しいリソース消費モデル



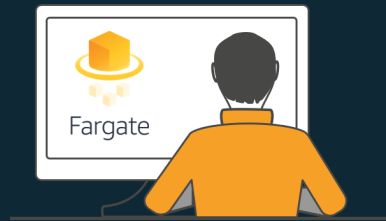
インスタンス
管理不要



タスク
ネイティブAPI



リソース
ベースの価格



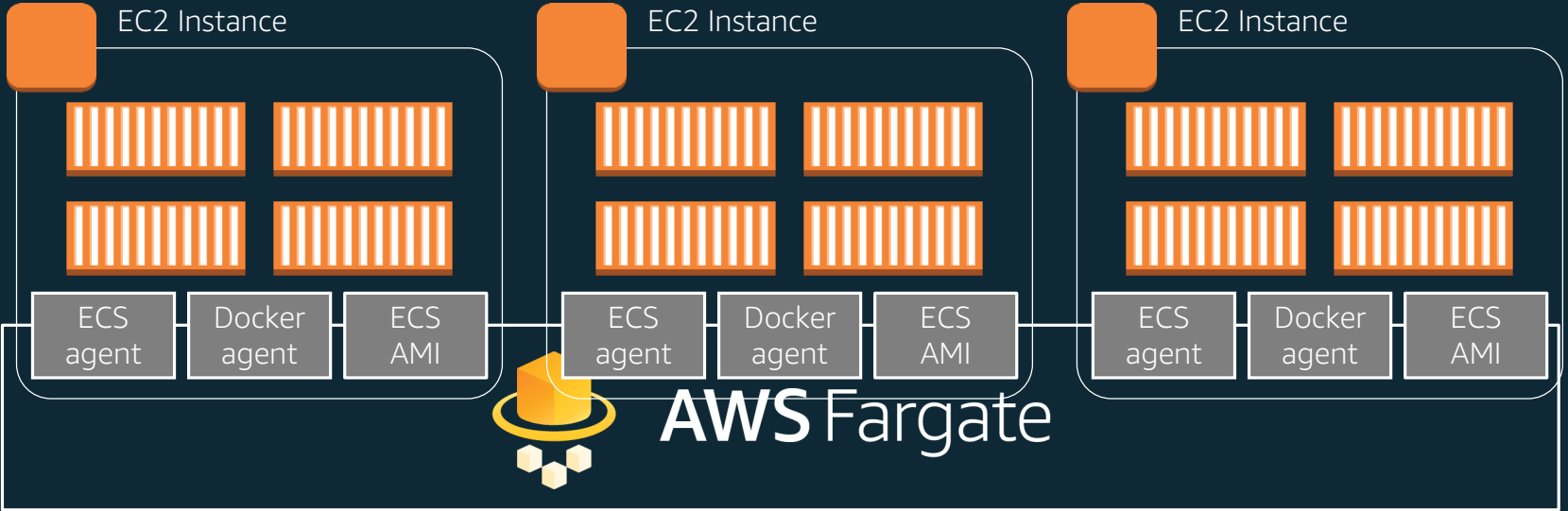
シンプルかつ強力で、使いやすい
新しいリソース消費モデル



Scheduling and Orchestration

Cluster Manager

Placement Engine



AWS Fargate

AWS上の本番環境のコンテナ運用を支援



Fargate



AWS VPCネットワークモード



タスク配置



他のAWSサービスとの深い連携



ECS CLI



グローバル展開



強力なスケジューラ



オートスケーリング



CloudWatch メトリクス



ロードバランサ

料金



課金基準: CPUとメモリ

秒単位で課金

タスクリソースに対する課金

- CPUとメモリはそれぞれの範囲内で独立に指定
-

```
{
  "cpu": "1 vCPU",
  "memory": "2 gb",
  "networkMode": "AWSVPC",
  "compatibilities": ["FARGATE",
    "EC2"],
  "placementConstraints": [],
  "containerDefinitions": [
    {
      <snip>.....
    }
  ]
}
```

Task Level Resources

タスクに割り当てるCPUとメモリの設定



柔軟な設定の選択肢

- 50 のCPU/メモリ設定から

CPU

Memory

256 (.25 vCPU)

512MB, 1GB, 2GB

512 (.5 vCPU)

1GB to 4GB (1GB 刻み)

1024 (1 vCPU)

2GB to 8GB (1GB 刻み)

2048 (2 vCPU)

4GB to 16GB (1GB 刻み)

4096 (4 vCPU)

8GB to 30GB (1GB 刻み)

Fargateでのコンテナ実行



Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.

FARGATE



Price based on task size

Requires network mode awsvpc

AWS-managed infrastructure, no Amazon EC2 instances
to manage

EC2



Price based on resource usage

Multiple network modes available

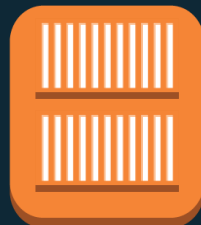
Self-managed infrastructure using Amazon EC2 instances

CONSTRUCTS



Task Definition

アプリケーションコンテナの定義
Image URL、CPU と Memory



Task

- コンテナ（群）の実行単位
- FARGATE launch type



Elastic Load
Balancer



Service

- ロングランニングアプリ用スケジューラ
- Taskの数を希望した数で維持
- ELBとの連携

Cluster

- インフラのアイソレーション単位
- Fargateの場合は論理的なアイソレーションの単位
- IAMパーミッションの境界

Fargateでのコンテナ実行



ECSと同一スキーマのTask Definition



ECS APIを使用した起動



容易な移行 - 同一クラスタ内でFargateとEC2の
ローンチタイプ

Networking

VPC INTEGRATION

AWS VPC Networking Mode

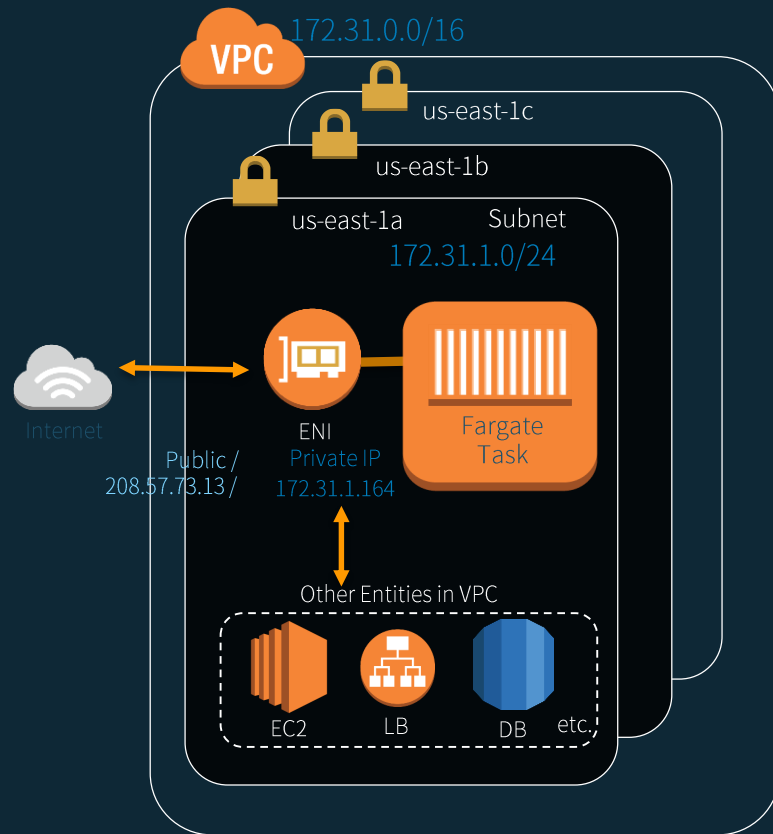
- 各タスクがネットワーク・インターフェースを持つ

FargateのタスクはユーザのVPC/サブネット内で実行

インバウンド/アウトバウンドのトラフィックを制御するためにセキュリティグループを構成

Public IPのサポート

複数AZ内のサブネットをまたがってアプリケーションを分散してレジリエンスを高める



インターネットアクセス

TaskとやりとりするすべてのネットワークトラフィックにTask ENIが利用される

Task ENIは以下の用途でも利用される

- ECRなどからのイメージのPull
- CloudWatchへのログ送付

アプリケーション自体がアウトバウンドのインターネットアクセスが不要でも、イメージのPullやログ送付のために必要

2種類のセットアップ方法

- アウトバウンドインターネットアクセスのみ可能なプライベートTask
 - インバウンドトラフィックは許可しない
- インバウンド/アウトバウンド両方のインターネットアクセスが可能なパブリックTask

LOAD BALANCING



ELBとの統合はserviceでサポート

Application Load BalancerとNetwork Load Balancerをサポート

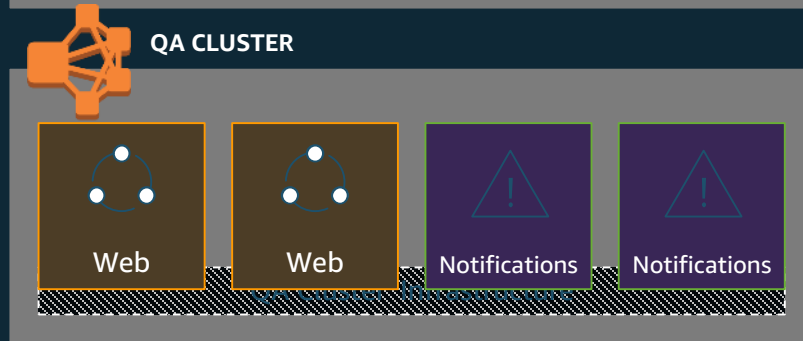
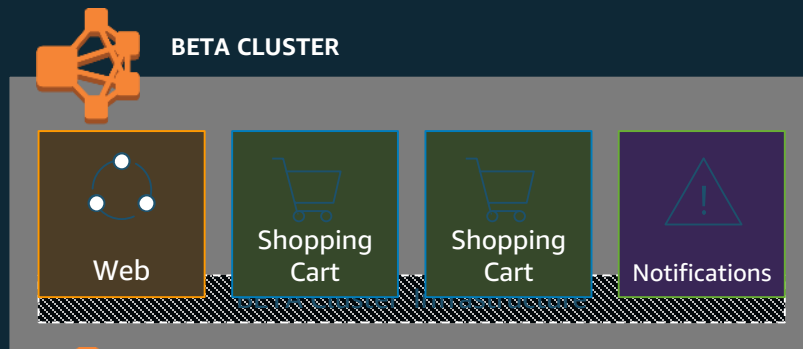
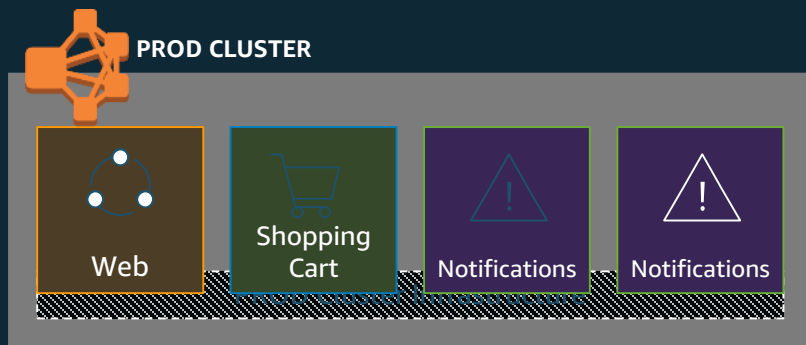
- Classic Load Balancerはサポートしない

ALBは最低2つのサブネット（異なるAZ）が必要

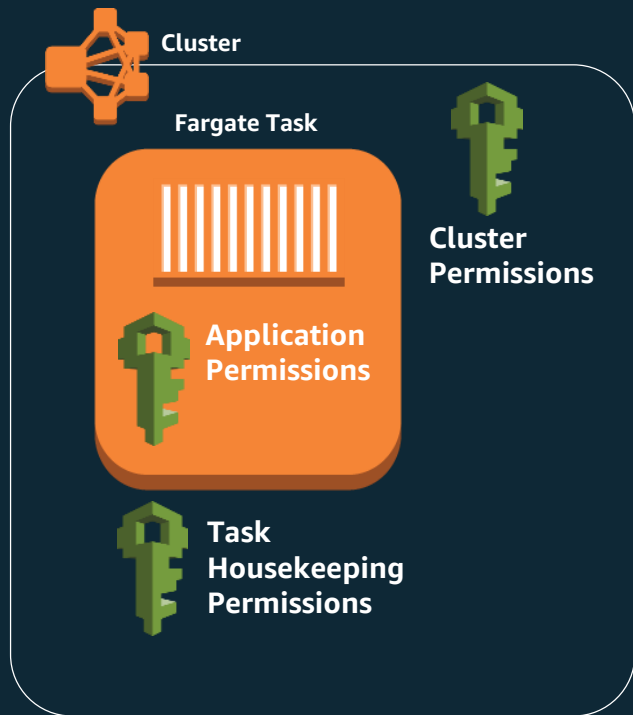
ALB/NLBのターゲットグループのtarget typeはipとする

SECURITY

クラスタレベルのアイソレーション



パーミッションのタイプ



Cluster Permissions:

誰がタスクを実行/参照できるか？

Application (Task) Permissions:

アプリケーションがアクセス可能なAWSリソースはどれか？

Housekeeping Permissions:

ECSに操作を許可したいパーミッションは何か？

e.g.

- ECR Image Pull
- CloudWatch Logs pushing
- ENI creation
- Register/Deregister targets into ELB

CONTAINER REGISTRIES



REGISTRY SUPPORT

Amazon Elastic Container Registry (ECR)



Public Repositories supported



3rd Party Private Repositories (coming soon!)



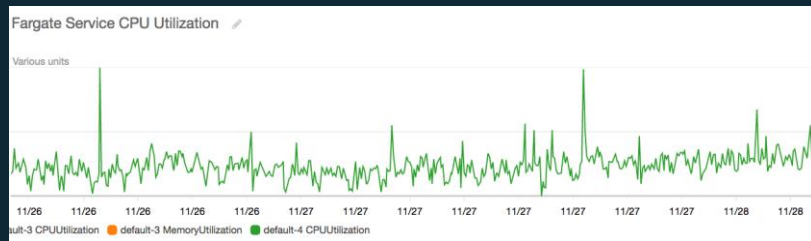
VISIBILITY AND MONITORING

CloudWatch Logs
CloudWatch Events



- Taskの状態変化

サービスレベルのCPU/Memory
使用量もCloudWatchで利用可能



STORAGE

Ephemeral storage backed by EBS

Container Storage Space – 10GB

タスク内のコンテナ用共有ボリュームスペース – 4GB



Fargateモード or EC2モード



Fargateでは難しいもの

- Windows Containers
- GPU Support
- docker execのようなインタラクティブなデバッグ
- SpotやRIベースの価格モデルの適用

これ以外はFargateでOK

FargateとLambdaの使い分け



AWS Lambdaを使うほうがいい場合

- イベントドリブン
- ミリ秒単位のコンピューート
- ランタイム管理をしたくない
- 分散バッチコンピューティング

それ以外はFargate

コンテナのAuto Scaling

コンテナの数をAuto Scalingさせる

何らかのメトリクスに応じて、コンテナの数を自動スケールさせたい

- コンテナのCPUやメモリ使用率、リクエスト数

コントロールプレーンの課題

- メトリクスの変化に対して、コンテナ数をどの程度変化させれば良い？

データプレーンの課題

- コンテナのスケールに応じて、インスタンス数もスケールが必要

→ ECSのTarget TrackingとFargateの組合せがオススメ

ECSの特徴: Target Trackingとの連携



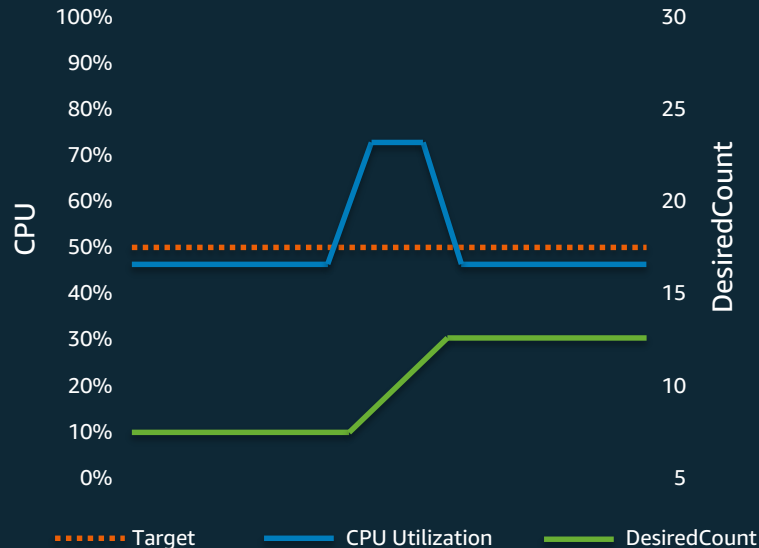
メトリクスに対してターゲットの値を設定するだけ

(例: CPU使用率 50%)

その値に近づく様に、Application Auto Scalingが自動的にServiceのDesiredCountを調整

ECSではコンソールからも設定可能

- ECSServiceAverageCPUUtilization
- ECSServiceAverageMemoryUtilization
- ALBRequestCountPerTarget



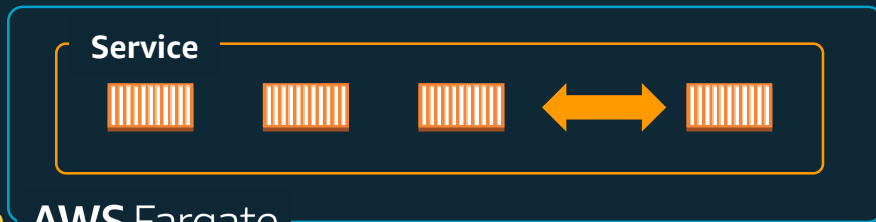
Fargateを利用したコンテナAuto Scalingの優位性

Fargateの場合

- Serviceのスケールに応じて自然にコンテナが起動・終了する
- コンテナの起動時間に対してのみ課金



AWS Fargate

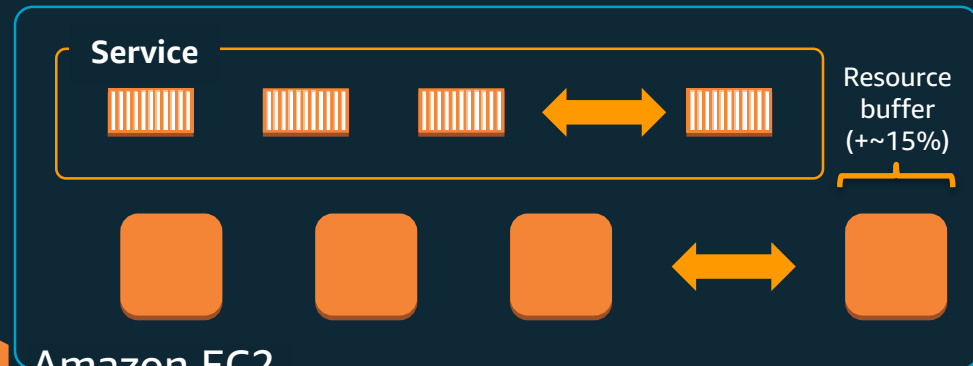


EC2の場合

- インスタンスのリソースも上手くスケールさせる必要があり煩雑
- 余分に持っているバッファ分もインスタンスの課金が必要



Amazon EC2





99.99%

EARLY ADOPTERS



PARTNER INTEGRATIONS



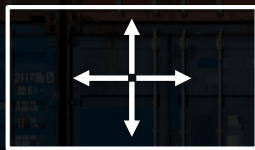


kubernetes

What is Kubernetes?



オープンソースの
コンテナ管理プラット
フォーム



コンテナの大規模な
運用に有用



モダンなアプリケーション
開発のための基本要素を提
供



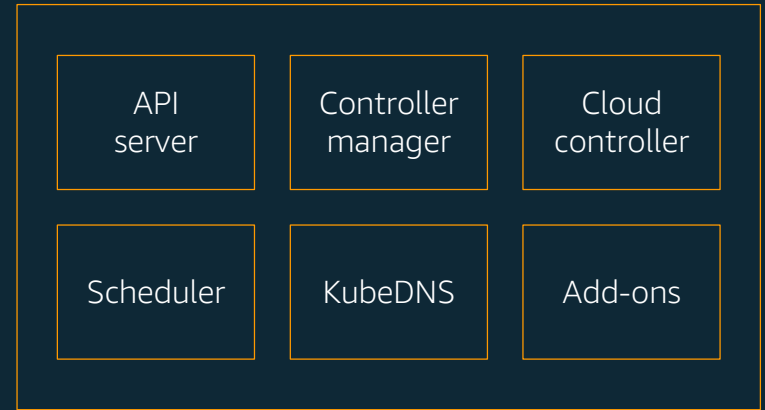
57% of Kubernetes workloads
run on AWS today
—CNCF survey

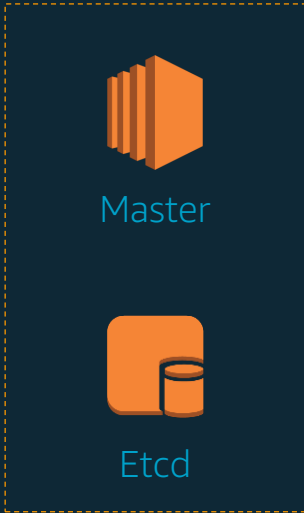
Kubernetes on AWS



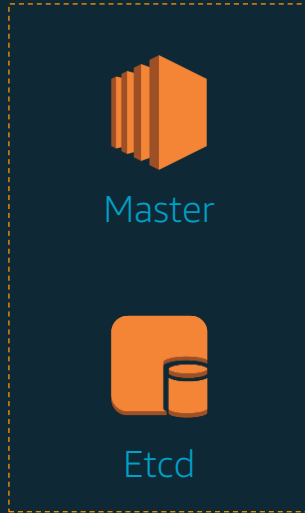
3x Kubernetes masters for HA

Kubernetes master

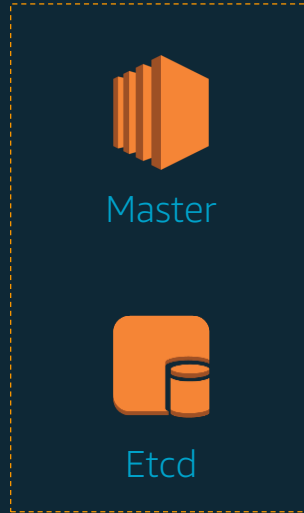




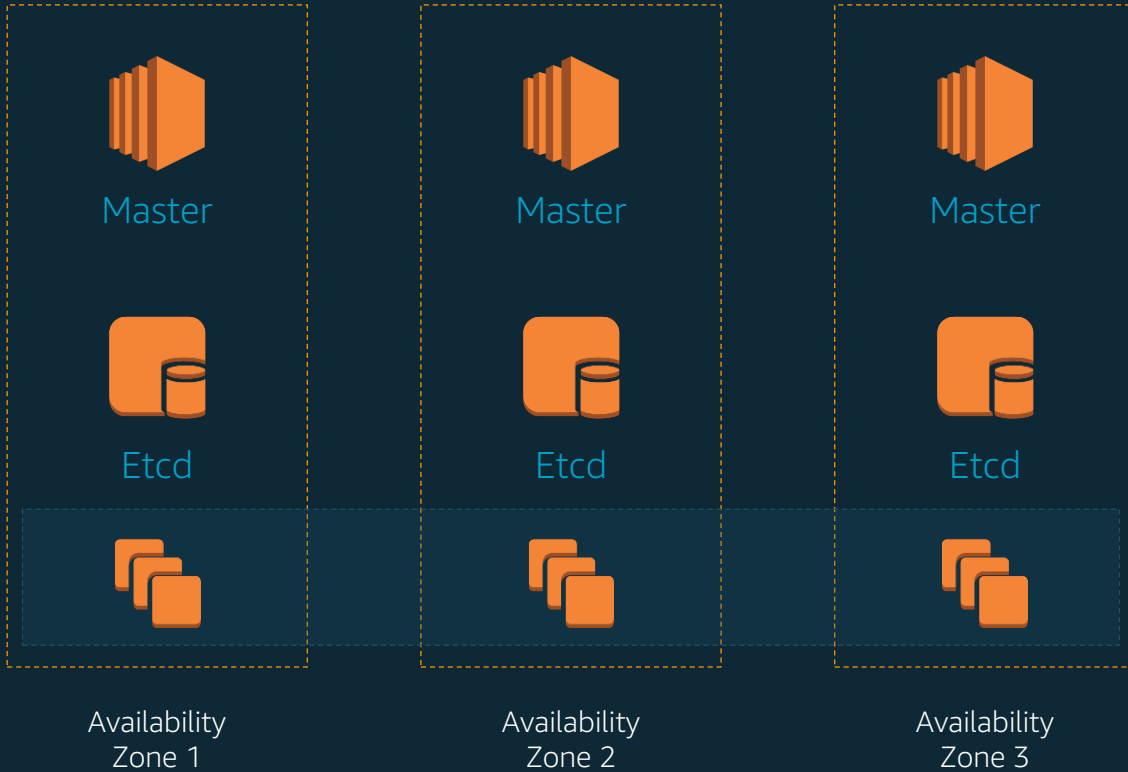
Availability
Zone 1



Availability
Zone 2



Availability
Zone 3





“Run Kubernetes for me.”



“Native AWS Integrations.”



"An Open Source Kubernetes Experience."



Amazon EKS

Amazon Elastic Container Service
for Kubernetes (EKS)

Tenet 1

EKSはエンタープライズ企業が本番のワークロードを実行するためのプラットフォームであること

Tenet 2

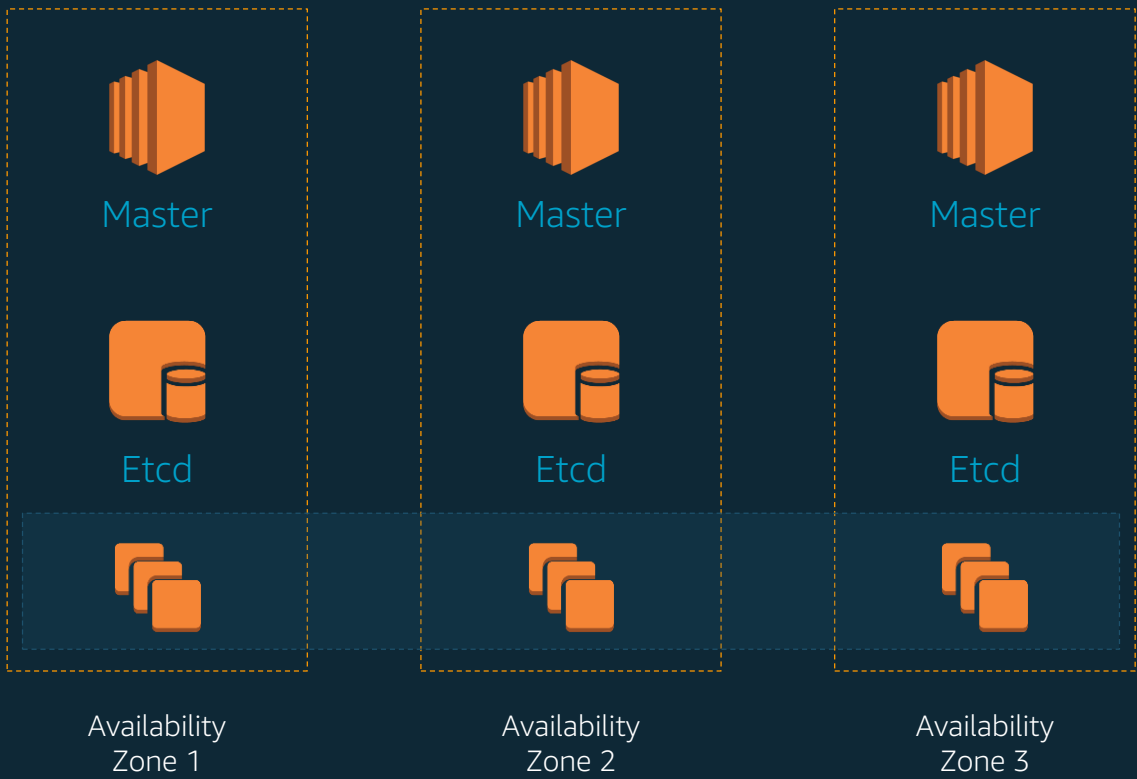
EKSはネイティブで最新のKubernetesの体験を提供すること

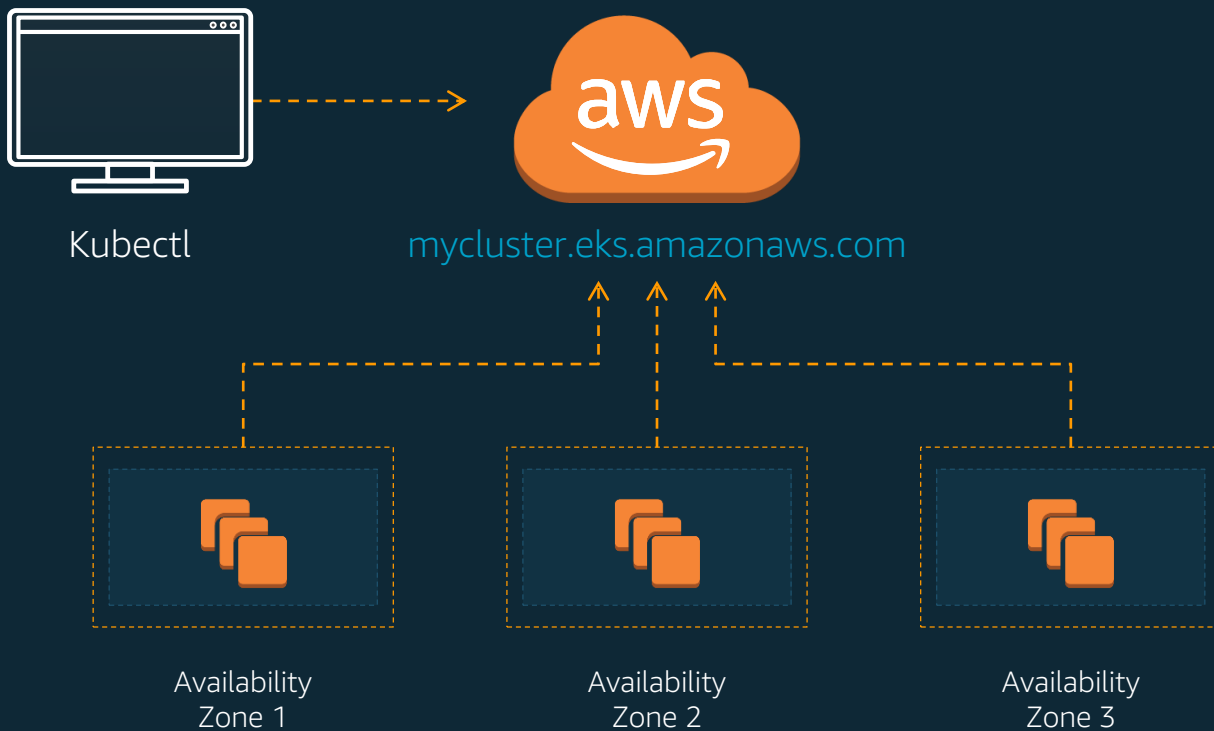
Tenet 3

EKSユーザが他のAWSサービスを使う時、シームレスな連携を実現し不要な作業を取り除くこと

Tenet 4

EKSチームは積極的にKubernetesプロジェクトに貢献していくこと





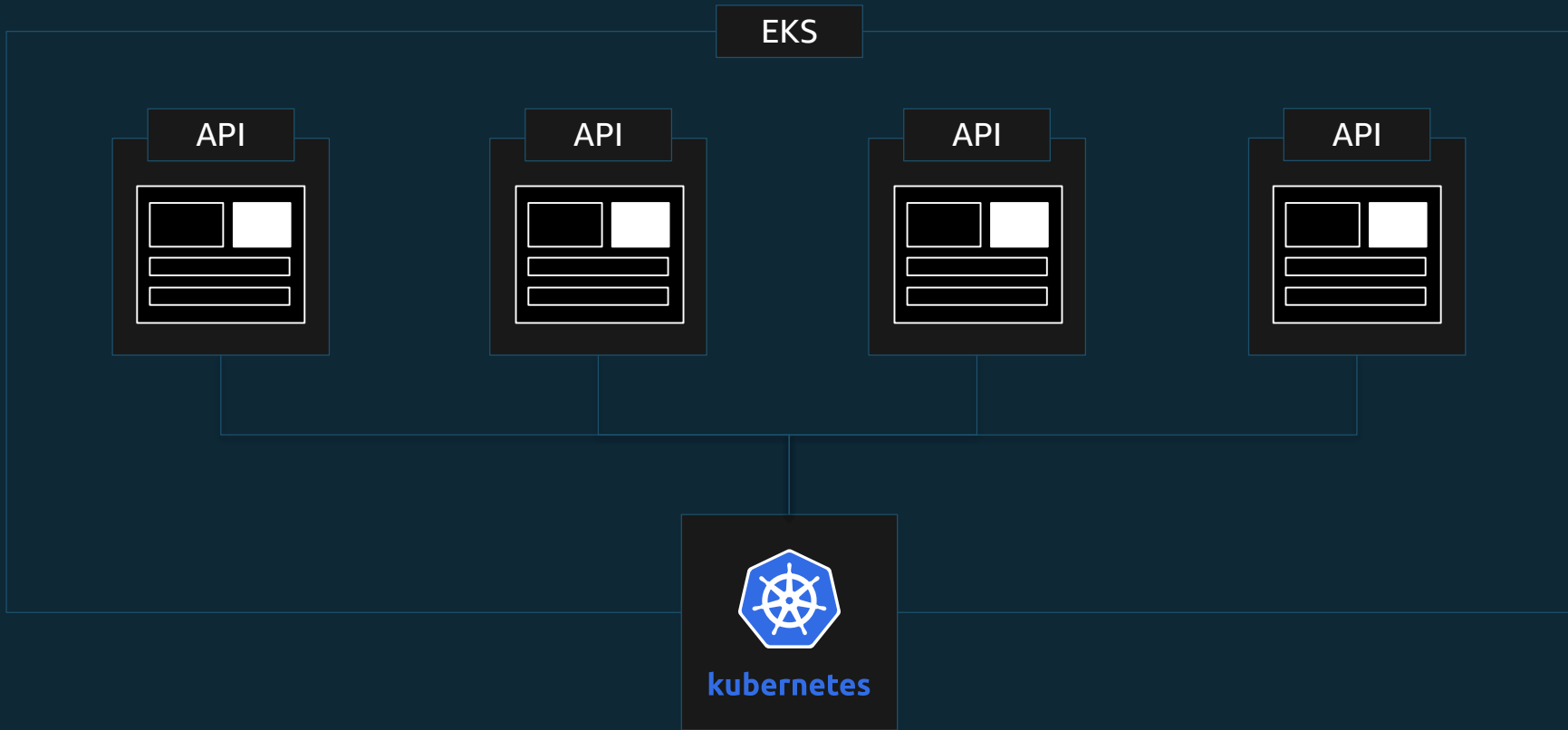
Kubernetes Certified



Kubernetes 準拠

1. ポータビリティと相互運用性を保証
2. タイムリーなアップデート
3. Confirmability

1.10 upstream == 1.10 in EKS



```
aws eks create-cluster --cluster-name reinvent2017 --desired-master-version  
1.7.1 --role-arn arn:aws:iam::account-id:role/role-name
```

aws eks create-cluster

HTTP/1.1 200 Content-type:
application/json

```
{ "cluster":  
  {  
    "clusterName": "string",  
    "createdAt": number,  
    "currentMasterVersion": "string",  
    "desiredMasterVersion": "string",  
    "masterEndpoint": "string",  
    "roleArn": "string",  
    "status": "string",  
    "statusMessage": "string"  
  }  
}
```

```
aws eks describe-cluster --cluster-name reinvent2017
```



```
aws eks describe-cluster -cluster-name reinvent2017
```

```
HTTP/1.1 200 Content-type:  
application/json
```

```
{ "cluster":  
  { "clusterName": "string",  
    "createdAt": number,  
    "currentMasterVersion": "string",  
    "desiredMasterVersion": "string",  
    "masterEndpoint": "string",  
    "roleArn": "string",  
    "status": "string",  
    "statusMessage": "string" }  
}
```

```
aws eks list-clusters
```

```
aws eks list-clusters
```

```
HTTP/1.1 200
```

```
Content-type: application/json
```

```
{  
  "clusterArns": [ "string" ],  
  "nextToken": "string"  
}
```

```
aws eks delete-cluster --cluster-name  
reinvent2017
```

```
aws eks delete-cluster --cluster-name reinvent2017
```

```
HTTP/1.1 200 Content-type:  
application/json
```

```
{ "cluster":  
  { "clusterName": "string",  
    "createdAt": number,  
    "currentMasterVersion": "string",  
    "desiredMasterVersion": "string",  
    "masterEndpoint": "string",  
    "roleArn": "string",  
    "status": "string",  
    "statusMessage": "string" }  
}
```

FargateのEKSサポート



Amazon ECS



Amazon EKS

An aerial photograph of a city grid, showing a dense arrangement of buildings and streets. The image is darkened with a semi-transparent black overlay, making the colors muted. The text "How is EKS architected?" is centered in white, sans-serif font.

How is EKS architected?

The background features a complex, abstract network of purple lines and dots, resembling a mesh or a data visualization, set against a dark background. The lines and dots are interconnected, creating a sense of depth and connectivity.


What is the master node configuration in EKS?

A close-up photograph of a person's hand pulling a server component out of a rack. The component is a dark, rectangular unit with a handle on the left side. The rack contains several other similar components, each with a green and yellow label. The background is dark and out of focus.

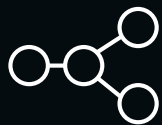
EKS Worker Nodes

How do I provision EKS Worker Nodes?



The background is a dark, futuristic digital environment. It features a glowing grid floor that recedes into the distance, creating a sense of depth. In the center, there is a large, glowing, semi-transparent sphere. The overall color palette is dark blue and black, with bright white and light blue highlights from the grid and sphere. The text is centered in the upper half of the image.

What is the networking configuration for EKS?



Native VPC networking
with CNI plugin



Pods have the same VPC
address inside the pod
as on the VPC



Simple, secure networking



Open source and
on Github

VPC CNI plugin

k8sとAWS VPC間をブリッジ

薄いレイヤのためパフォーマンスインパクトは極小

Pod IP ← ENI secondary IP

How do I use it?

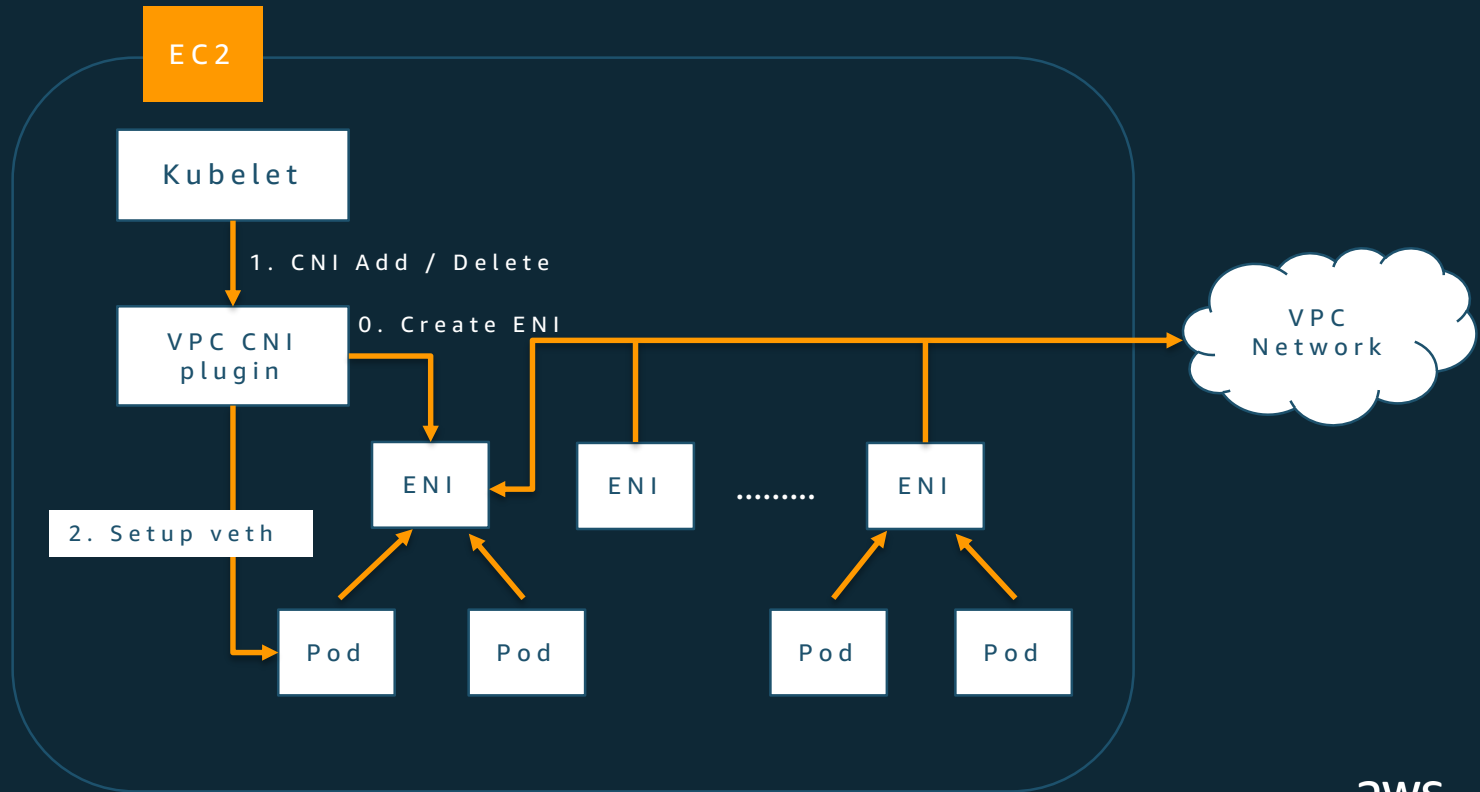
AWS上のk8sであれば利用可能

- EKS
- 自前でAWS上に構築したk8s

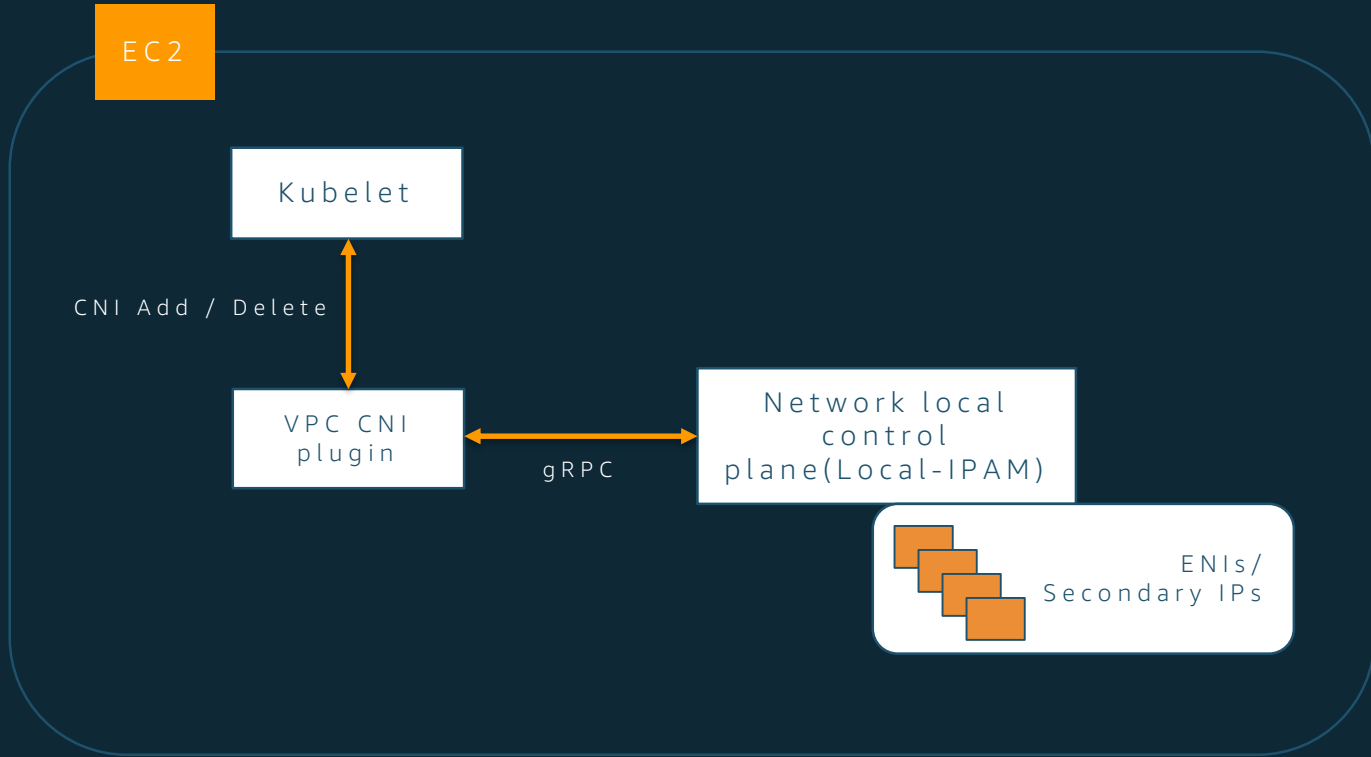
Daemonset としてデプロイ.

```
kubectl create -f eks-cni.yaml
```

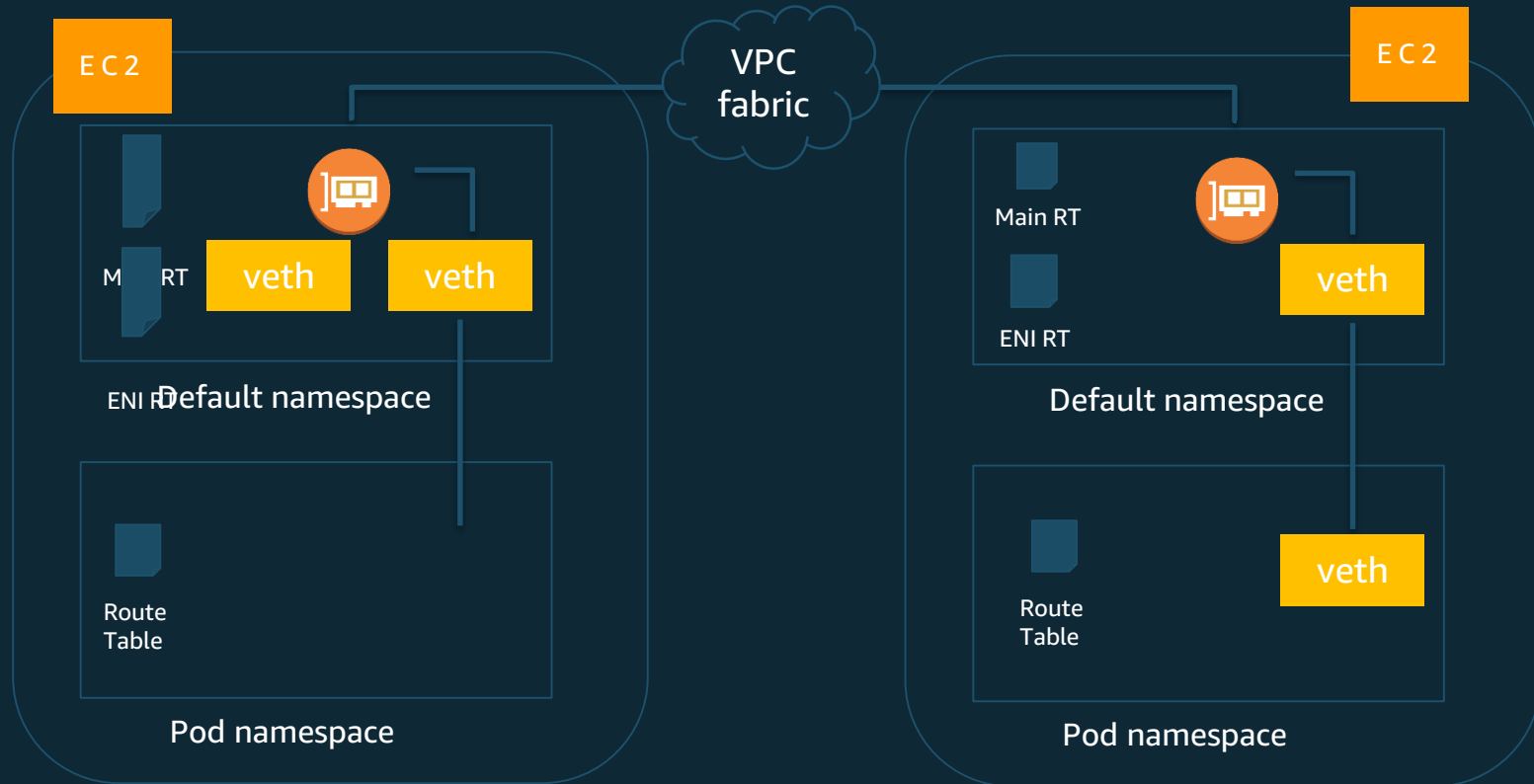
VPC CNI networking internals



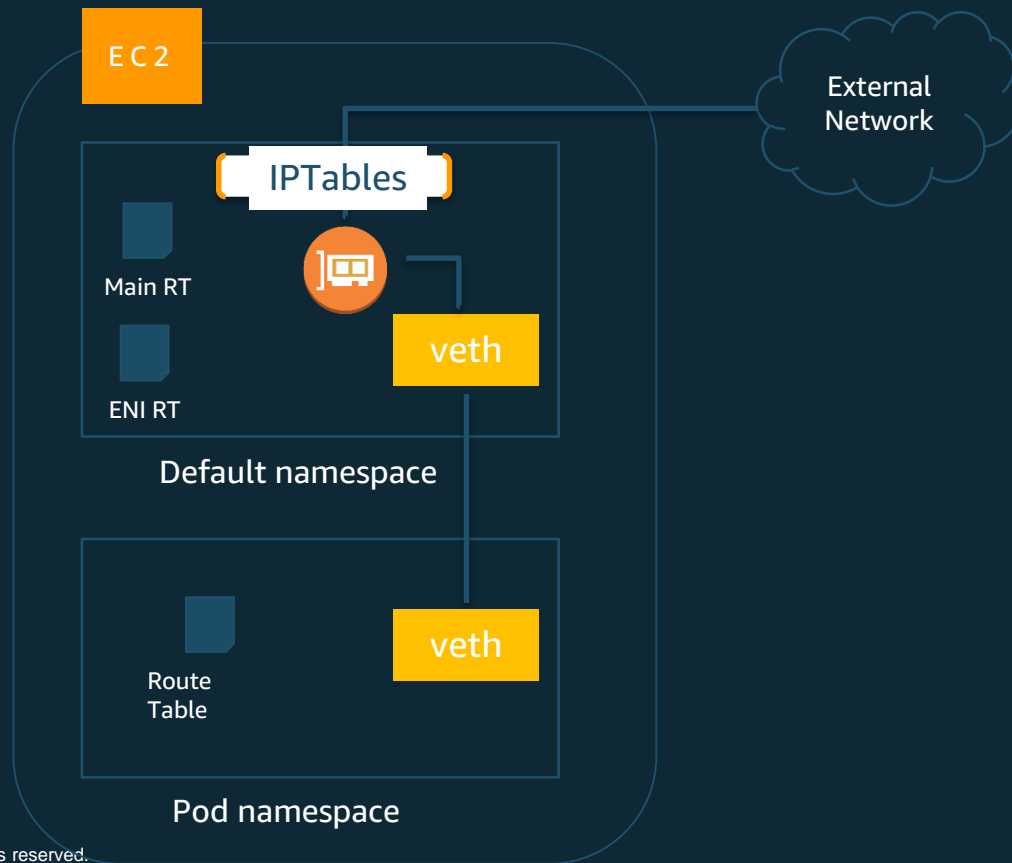
VPC CNI plugin architecture



PodとPodのパケットの流れ



Podと外部のパケットの流れ



Load Balancing

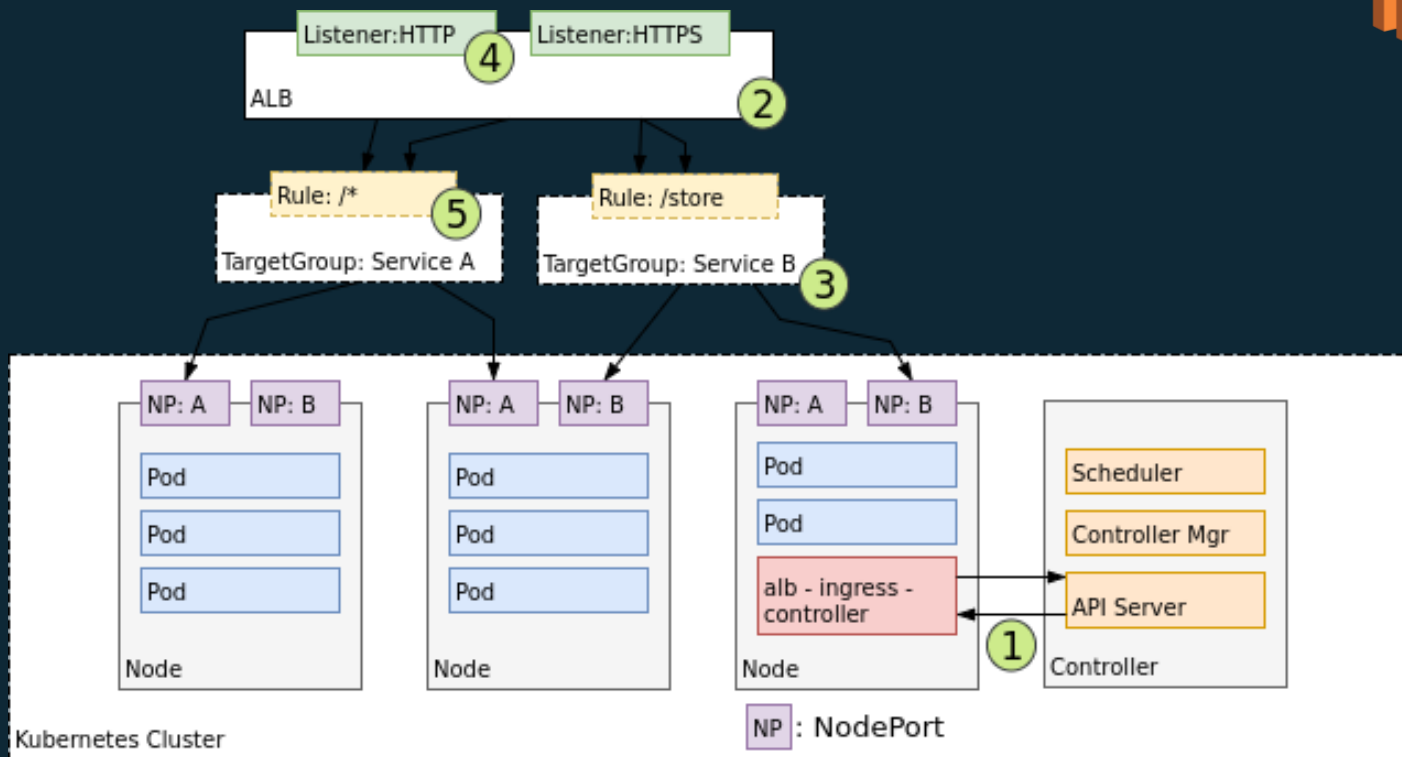


CoreOS ALB Ingress Controller: Supported by AWS

IngressリソースとしてALBを公開できる

ホスト/パスによるcontent-baseルーティングをサポートしたL7の負荷分散が可能

Load Balancing



Load Balancing



Network Load Balancer: 1.9以降サポート (Alpha)

L4ロードバランサ、service typeとして“LoadBalancer”を指定

多くのケースでClassic Load Balancerの置換えとなり得る

- 現状、LoadBalancerを指定するとClassic Load Balancerが作成される

A blurred background of server racks in a data center, with light trails suggesting motion or data flow.

How do I configure network security with EKS?



Kubernetes Network Policies enforce network security rules



Calico is the leading implementation of the network policy API



Open source, active development (>100 contributors)



Commercial support available from Tigera



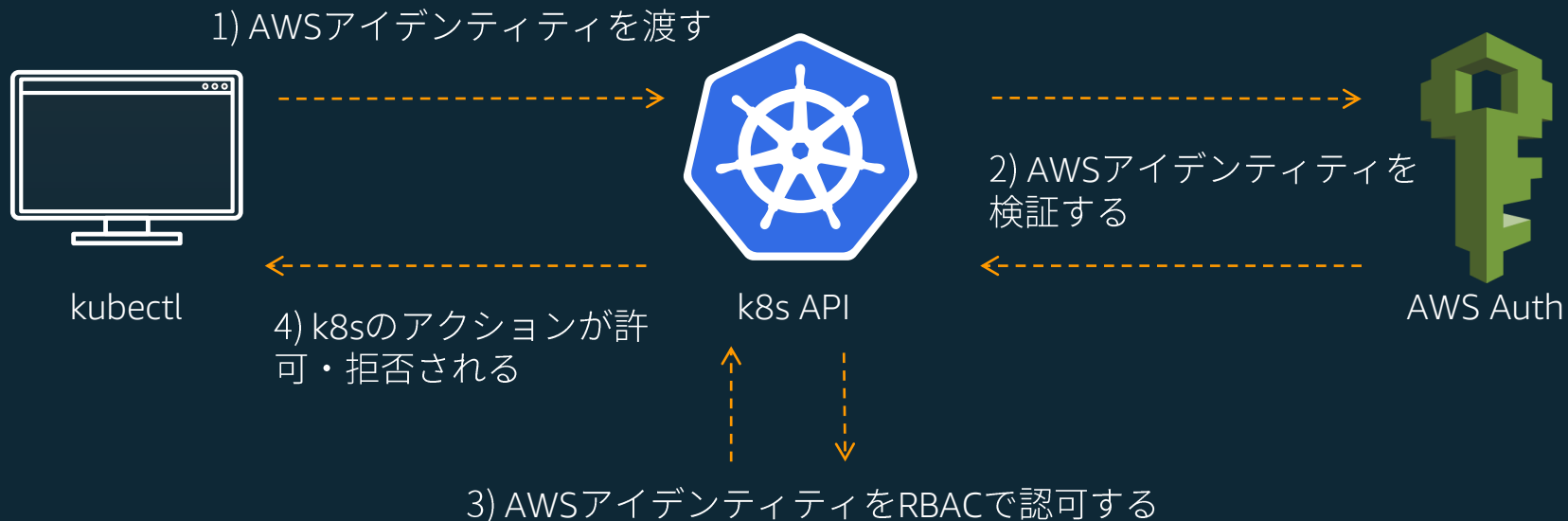
How does IAM authentication work with Kubernetes?

Heptio IAM Authenticator

An open source approach to integrating
AWS IAM authentication with Kubernetes



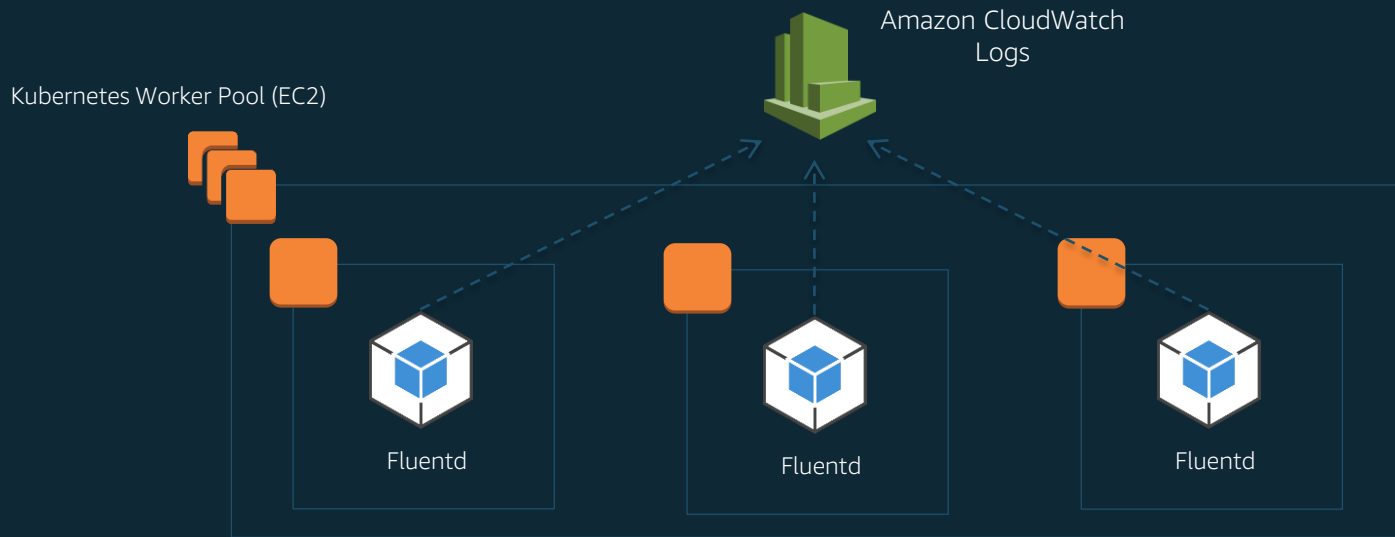
IAM認証 + kubectl



How do I get visibility
into my EKS Masters?

```
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
elif_operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True
#selection at the end -add back the deselected mirror modifier object
if mirror_ob.select == 1:
    modifier_ob.select = 1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
mirror_ob.select = 0
time = bpy.context.selected_objects[0]
bpy.data.objects[mirror_ob.name].select = 1
```

Fluentdを用いたCloudWatch Logsへのログ集約



<https://github.com/fluent/fluentd-kubernetes-daemonset>

Metrics

Visualizer

Grafana, Kibana, Dashboard

Alerting

AlertManager, Kapacitor

Cluster-wide Aggregator

Prometheus, Heapster

Nodes

Node exporter

Pod/Container

Kube-state-metrics
cAdvisor

Application

/metrics
JMX

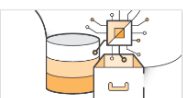



Data Model

InfluxDB, Graphite

オンラインセミナー資料の配置場所

AWS クラウドサービス活用資料集

- <https://aws.amazon.com/jp/aws-jp-introduction/>

			
サービス別資料	ソリューション別資料	業種別資料	その他の資料
無料オンラインセミナー「Black Belt Online Seminar」のサービスカット資料他、AWSのTechメンバーによる各サービスの解説資料がご覧いただけます。	無料オンラインセミナー「Black Belt Online Seminar」のソリューションカット資料他、特定のソリューションについてのAWS活用方法がご覧いただけます。	無料オンラインセミナー「Black Belt Online Seminar」のインダストリーカット資料他、特定の業界のユースケースがご覧いただけます。	イベントに関する資料やアップデート情報などをご覧いただけます。

Amazon Web Services ブログ

- 最新の情報、セミナー中のQ&A等が掲載されています。
- <https://aws.amazon.com/jp/blogs/news/>

公式Twitter/Facebook AWSの最新情報をお届けします



@awscloud_jp



検索

もしくは

<http://on.fb.me/1vR8yWm>

最新技術情報、イベント情報、お役立ち情報、
お得なキャンペーン情報などを日々更新しています！

AWSの導入、お問い合わせのご相談

AWSクラウド導入に関するご質問、お見積、資料請求をご希望のお客様は以下のリンクよりお気軽にご相談下さい。

<https://aws.amazon.com/jp/contact-us/aws-sales/>



The screenshot shows the AWS contact form for Japan. On the left is a navigation menu with the following items: 'お問い合わせ' (Contact Us), '日本担当チームへのお問い合わせ' (Contact Us to the Japan Support Team) with a right-pointing arrow, '関連リンク' (Related Links), and 'フォーラム' (Forum). The main content area is titled '日本担当チームへのお問い合わせ' in orange. Below the title, there is a paragraph of text: 'AWS クラウド導入に関するご質問、お見積り、資料請求をご希望のお客様は、以下のフォームよりお気軽にご相談ください。平日営業時間内に日本オフィス担当者よりご連絡させていただきます。' (For customers who have questions, requests for quotes, or request materials for AWS cloud migration, please contact us via the form below. We will contact you from our Japan office staff during business hours on weekdays.) Below this is a bolded note: '※ご請求金額またはアカウントに関する質問はこちらからお問い合わせください。' (Please contact us here for questions regarding billing amounts or accounts.) followed by another note: '※Amazon.com または Kindle のサポートにお問い合わせはこちらからお問い合わせください。' (Please contact us here for support regarding Amazon.com or Kindle.) Below these notes is a warning: 'アスタリスク (*) は必須情報となります。' (Asterisk (*) indicates required information.) At the bottom, there are two input fields: '姓*' (Last Name) and '名*' (First Name), both with asterisks indicating they are required. The '姓*' field has a small 'id' icon on the right side.

※ 「AWS お問い合わせ」で検索して下さい。

AWS Well Architected 個別技術相談会お知らせ

- Well Architectedフレームワークに基づく数十個の質問項目を元に、お客様がAWS上で構築するシステムに潜むリスクやその回避方法をお伝えする個別相談会です。

<https://pages.awscloud.com/well-architected-consulting-jp.html>

- 参加無料
- 毎週火曜・木曜開催

【毎週火、木曜開催】AWS Well-Architected 個別技術相談会

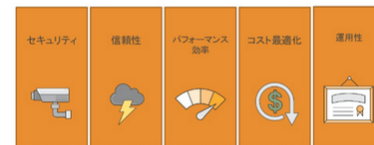
AWS 上で構築するシステムのリスクの把握・回避方法をご希望のお客様

この度 AWS をご活用頂いているお客様を対象に「AWS Well-Architected 個別技術相談会」を開催致します。

Well-Architected 個別技術相談会では、リスクの把握・回避を目的として、セキュリティ・信頼性・パフォーマンス・コスト・運用の5つの観点で、お客様の AWS 活用状況や構成についてお伺いします。AWS のベストプラクティスに基づき作成された Well-Architected フレームワークを元に、今までお客様がお気づきでなかったリスクやAWS活用の改善点を見つけることができます。例えば、自動車においては納車前点検、車検を定期的に行うのと同様に、本相談会はおお客様の AWS 上のシステムをよりよく活用頂くことを目的としております。

» [説明資料\(PDF\)](#) [AWS Well-Architected Framework -クラウド設計・運用ベストプラクティスの活用-]

Well-Architected 個別技術相談会にご参加頂くには、本ページにてお申込み後、弊社担当者からお送りするヒアリングシートにご記入・担当者にご送付頂く必要があります。その内容を元に、当日の相談会では AWS のソリューションアーキテクトと共に技術的なディスカッションをさせて頂きます。また、遠方のお客様、アマゾン東京オフィスへのご来社が時間等の関係で難しいお客様は、Web のプレゼンテーションツールや、お電話を活用したリモートでのご相談も承ります。



下記のフォームよりお申込みください。

* 姓:

* 名: